OPERATING SYSTEM RESOURCE PROTECTION

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Application No. 60/513,941, filed Oct. 24, 2003. Filed simultaneously herewith is U.S. non-provisional patent application entitled "Application Identity for Software Products," attorney docket number MS#307048.01 (5102) (which also claims the benefit of U.S. Provisional Application No. 60/513,941, filed Oct. 24, 2003), the entire disclosure of which is incorporated herein by reference.

TECHNICAL FIELD

[0002] Embodiments of the present invention relate to the field of operating systems for computers. In particular, embodiments of this invention relate to managing the installation, execution, and removal of applications by an operating system.

BACKGROUND OF THE INVENTION

[0003] While operating systems have made dramatic strides in improving their usability and reliability, the user experience relating to the installation, management, and removal (i.e., uninstallation) of application programs still needs improvement. For example, an application program may incorrectly configure a system setting during installation or overwrite a file needed by another application program. It may also be difficult for users to uninstall undesirable applications such as ad-ware and spy-ware. Many system crashes and performance degradation (e.g., slow boot times) may also be attributable to application problems. For example, the following situations may cause an application program and possibly the underlying operating system to fail: an incomplete uninstall of an application, over deletion when uninstalling an application program, and improperly stored files.

[0004] In some current operating systems, a newly installed application program may overwrite a shared dynamic-link library (DLL) file with an older or newer version needed by the newly installed application program. If the older or newer file is incompatible with the overwritten file, a currently installed application program dependent on the overwritten file may crash when attempting to access the overwritten file.

[0005] Accordingly, an improved system and method for managing application impact is desired to address one or more of these and other disadvantages.

SUMMARY OF THE INVENTION

[0006] Embodiments of the invention include a method for allowing an operating system to protect its resources. In an embodiment, the invention includes employing a persistent, individual identity associated with an application program or a group of application programs to allow an operating system to identify and differentiate between the different application programs or groups of application programs and components thereof.

[0007] The operating system or other program manipulates the application programs via the identities associated with each of the application programs. For example, the operating system uses the identities to (1) ensure clean uninstalls, (2) prevent an application from accessing a service or performing an action for which the application does not have authorization, (3) virtualize system resources to better isolate applications from each other, (4) enable application impact rollback (e.g., revert file type associations to a pre-application install state), and (5) enable file and registry ownership tracking. Protection mechanisms include, but are not limited to, providing read-only access, logging changes to enable rollback, and virtualizing resources per application and per user. For example, the operating system generates a copy of a write-protected file for an application program that has requested write access to the write-protected file.

[0008] In accordance with one aspect of the invention, a method grants an application program access to a resource on a computing system. The method includes receiving a request from an application program for access to a resource identified in the request. The method also includes determining an application identifier for the application program. The method includes identifying a privilege from a manifest as a function of the determined application identifier and the identified resource. The manifest indicates the privilege that the application program has for accessing the identified resource. The method also includes granting the application program access to the identified resource according to the identified privilege.

[0009] In accordance with another aspect of the invention, one or more computer-readable media have computer-executable components for granting an application program access to a resource. The components include an interface module to receive a request from an application program for access to a resource identified in the request. The components also include an identity module to determine an application identifier for the application program to distinguish the application program and components thereof from other application programs. The components also include a filter module to identify a privilege from a manifest as a function of the application identifier determined by the identity module and the identified resource. The manifest indicates the privilege that the application program has for accessing the identified resource. The components also include an access control module to grant the application program access to the identified resource according to the privilege identified by the filter module.

[0010] In accordance with yet another aspect of the invention, a computer-readable medium stores a data structure that represents a manifest specifying access rights of an application program to access a plurality of resources. The data structure includes a first field storing a value representing an identity corresponding to the application program. The data structure also includes a second field storing a list of resources associated with the application program. The data structure further includes a third field storing a privilege associated with the identity from the first field and with the list of resources stored in the second field. The privilege defines an access right of the application program to access each resource in the list of resources.

[0011] In accordance with still another aspect of the invention, a system grants an application access to a system resource. The system includes a memory area to store a manifest. The manifest maps an application identifier and a